

Preparing for the Exercises

Description

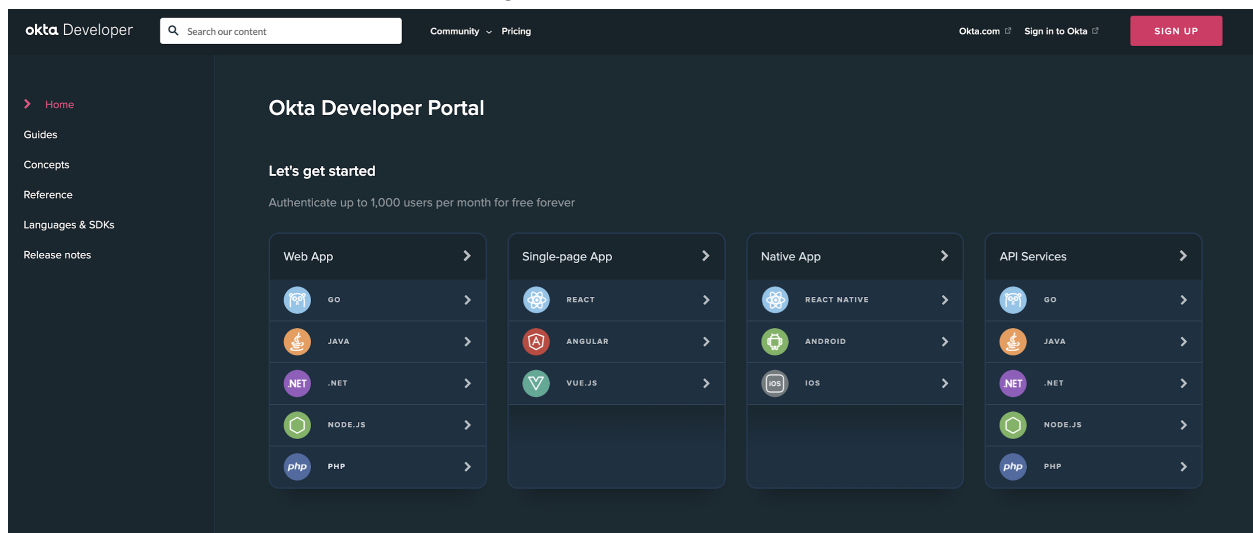
In this exercise you'll sign up for a developer account at developer.okta.com to follow along with the exercises. Once you have an account, you'll be able to create applications and configure your OAuth server to follow along with the assignments.

Estimated Duration

10 minutes

Instructions

Please visit developer.okta.com and sign up for an account.



Once you've registered, navigate to **Security** then click **API** in the side menu. On that page, you'll see a list of your authorization servers. There should be only one, called "default".

The screenshot shows the Okta Admin Console interface. On the left is a navigation sidebar with categories like Dashboard, Directory, Applications, and Security. The main content area is titled 'API' and has tabs for 'Authorization Servers', 'Tokens', and 'Trusted Origins'. The 'Authorization Servers' tab is active, displaying a table with the following data:

Name	Audience	Issuer URI	Status	Actions
default	api://default	https://dev-7533118.okta.com/oauth2/default	Active	[Edit]

Below the table is a 'Show More' link. At the top of the table area is a search bar and an 'Add Authorization Server' button. The footer contains copyright information, version details, and links to download the Okta plugin and provide feedback.

Copy the **Issuer URI** from that list, that is the identifier of your OAuth server and you'll need it throughout the course.

Next, we need to create a scope that the application can request from your API. We'll talk more about what scopes are for and how they are used in later lessons, but we need one in order to be able to complete a flow for now.

Click on the name **default** to view the details of your OAuth server. Then click on the **Scopes** tab at the top.

default

[Help](#)

Active ▾

Settings Scopes Claims Access Policies Token Preview

+ Add Scope					
Name	Display Name	Description	User Consent	Default Scope	Metadata Publish
openid	openid	Signals that a request is an OpenID request.	No	No	Yes ✎
profile	View your profile information.	The exact data varies based on what profile information you have provided, such as: name, time zone, picture, or birthday.	No	No	Yes ✎
email	View your email address.	This allows the app to view your email address.	No	No	Yes ✎
address	View your address.	This allows the app to view your address, such as: street address, city, state, and zip code.	No	No	Yes ✎
phone	View your phone number.	This allows the app to view your phone number.	No	No	Yes ✎
offline_access	Keep you signed in to the app.	This keeps you signed in to the app, even when you are not using it.	No	No	Yes ✎

The server is preconfigured with OpenID Connect scopes and the `offline_access` scope. For the exercises, you'll need to add your own custom scope to get an access token with that scope. It doesn't matter what it's called for now, so use a word that represents the API you're building, like "photos".

Click on **Add Scope**, and enter the name of the scope in the first field. It's usually a good idea to stick to lowercase letters. Make sure to also click the **"Include in public metadata"** checkbox at the bottom as well.

Add Scope

Name

photos

For example: email

Display phrase

Access your photos

For example: Access your email

22 characters remaining

Description

For example: This allows you to use your email to login to the app

User consent

☐ Require user consent for this scope

Default scope

☐ Set as a default scope

Metadata

☒ Include in public metadata

Create

Cancel

Once you've created the custom scope you're ready to complete the exercise. To check your work, we'll be using a companion tool for this course which lives at oauth.school.

Visit the website oauth.school, and in the first “Getting Started” exercise, paste the issuer URI. This will check that it can find your OAuth server and find your custom scope you added in the previous step. If everything worked, the website will show you the custom scope it found.

Getting Started

In this exercise you'll create a new Okta developer account and enter your Issuer URL. Please make sure to also add at least one custom scope and make that scope available in your server's public metadata.

Great! Your issuer URL is accepted and we found 1 custom scopes!

Issuer URL

https://dev-7533118.okta.com/oauth2/default

We'll save the issuer URL to use it when checking your work in the following exercises

Scopes

photos

We found the following custom scopes in your OAuth server metadata

Next you'll need to find the server's authorization endpoint and token endpoint for use in the later exercises. These can be found programmatically by fetching the server's metadata URL.

Back in your list of authorization servers, choose your default server and click the **Settings** tab.

default

[Help](#)

Active ▾

Settings Scopes Claims Access Policies Token Preview

Settings

[Edit](#)

Name	default
Audience	api://default
Description	Default Authorization Server for your Applications
Issuer	https://dev-22588100.okta.com/oauth2/default
Metadata URI	https://dev-22588100.okta.com/oauth2/default/.well-known/oauth-authorization-server
Signing Key Rotation ⓘ	Automatic
Last Rotation	5 Mar 2021


Authorization Servers

An authorization server defines your security boundary, and is used to mint access and identity tokens for use with OIDC clients and OAuth 2.0 service accounts when accessing your resources via API. Within each authorization server you can define your own OAuth scopes, claims, and access policies. Read more at [help page](#)

One of the fields visible is your server's Metadata URI. You might notice that it's actually based off of your server's Issuer URI. That is described in the OAuth Server Metadata extension, which

says that the metadata URI should be created by appending the .well-known path to the issuer URI. If you click on that link, you'll see that it's actually a JSON file with a bunch of properties that describe the server.

(Note: It is helpful to install a browser plugin that can format JSON into a nice tree!)



```
<  →  ↻  🔒 dev-7533118.okta.com/oauth2/default/.well-known/oauth-authorization-server

{
  "issuer": "https://dev-7533118.okta.com/oauth2/default",
  "authorization_endpoint": "https://dev-7533118.okta.com/oauth2/default/v1/authorize",
  "token_endpoint": "https://dev-7533118.okta.com/oauth2/default/v1/token",
  "registration_endpoint": "https://dev-7533118.okta.com/oauth2/v1/clients",
  "jwks_uri": "https://dev-7533118.okta.com/oauth2/default/v1/keys",
  "response_types_supported": [
    "code",
    "token",
    "id_token",
    "code id_token",
    "code token",
    "id_token token",
    "code id_token token"
  ],
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post",
    "okta_post_message"
  ],
  "grant_types_supported": [
```

Most of these properties we don't care about right now. We really only care about the `authorization_endpoint` and `token_endpoint` values. These two happen to also be prefixed with the Issuer URI, but that isn't something you should rely on, which is why it's important to fetch the metadata URL and grab the actual values from there.

Copy these two values somewhere you'll be able to find them again when you start the exercises. You'll need these throughout the course.

Check your work by entering the authorization endpoint and token endpoint URLs into the website.

Great! Your issuer URL is accepted and we found 1 custom scopes!

Issuer URL

`https://dev-7533118.okta.com/oauth2/default`

We'll save the issuer URL to use it when checking your work in the following exercises

Scopes

`photos`

We found the following custom scopes in your OAuth server metadata

Authorization Endpoint

Find your server's authorization endpoint and enter it here

Token Endpoint

Find your server's token endpoint and enter it here

Check

Once you've entered them correctly, you'll be able to continue on with the next exercises.